

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-063873  
 (43)Date of publication of application : 06.03.1998

(51)Int.CI. G06T 15/00

(21)Application number : 08-222571 (71)Applicant : ATR TSUSHIN SYST

KENKYUSHO:KK

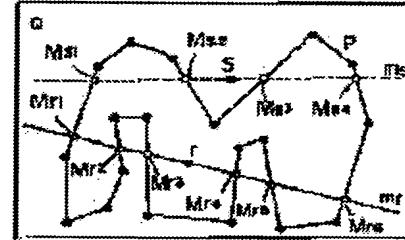
(22)Date of filing : 23.08.1996 (72)Inventor : KITAMURA YOSHIFUMI  
 KISHINO FUMIO

## (54) OCTAL TREE GENERATING METHOD

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To generate the octal tree representation of a body in a nonprojection shape fast from polygon representation by generating a polygonally-sectioned voxel with some plane of a three-dimensional body and generating the octal tree by using a voxel decided to have a representative point inside.

**SOLUTION:** A straight line Mr crosses the outline (r) or an (n)-gon at Mr1-Mr3 on one side and at Mr4-Mr6 on the other side each even-number of times, so it is judged that the point (r) is in an (n)-gon P. A straight line Ms passing a point (s) in a plane Q, on the other hand, crosses the outline of the (n)-gon P at Ms1 and Ms2 on one side of the point (s) and at Ms3 and Ms4 on the other hand each odd-number of times T, so it is judged that the point (s) is outside the (n)-gon. A voxel having the point (r) inside is called a black voxel and a voxel having it outside is called a white voxel. Then an octal tree is generated by using voxels having representative points inside. Consequently, this method is applicable even to a nonprojection body and the octal tree can be generated fast from polygon representation.



## LEGAL STATUS

[Date of request for examination] 23.08.1996

[Date of sending the examiner's decision of rejection] 24.07.2001

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision]

of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-63873

(43)公開日 平成10年(1998)3月6日

(51)Int.Cl.<sup>6</sup>  
G 0 6 T 15/00

識別記号 庁内整理番号

F I  
G 0 6 F 15/72

技術表示箇所

4 5 0 A

審査請求 有 請求項の数 3 O.L (全 5 頁)

(21)出願番号 特願平8-222571

(22)出願日 平成8年(1996)8月23日

(71)出願人 000127695

株式会社エイ・ティ・アール通信システム  
研究所  
京都府相楽郡精華町大字乾谷小字三平谷5  
番地

(72)発明者 北村 喜文

京都府相楽郡精華町大字乾谷小字三平谷5  
番地 株式会社エイ・ティ・アール通信シ  
ステム研究所内

(72)発明者 岸野 文郎

京都府相楽郡精華町大字乾谷小字三平谷5  
番地 株式会社エイ・ティ・アール通信シ  
ステム研究所内

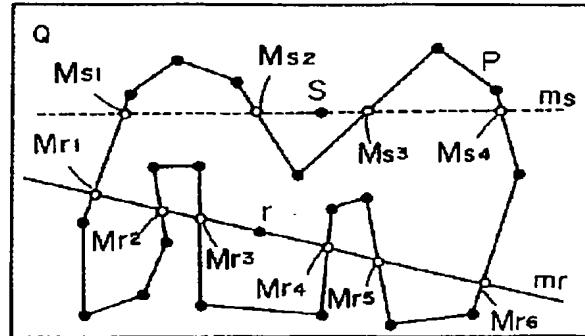
(74)代理人 弁理士 深見 久郎 (外2名)

(54)【発明の名称】 八分木生成方法

(57)【要約】

【課題】 非凸形状の物体の八分木表現を多面体表現から高速に作成できるような八分木作成方法を提供する。

【解決手段】 三次元物体1のある平面Qで断面多角形のボクセルを生成し、断面多角形の内部に代表点があるか否かを判別し、内部に代表点のあることが判別されたボクセルを用いて八分木を生成する。



## 【特許請求の範囲】

【請求項1】三次元物体の多面体表現から八分木を生成する八分木生成方法であって、前記三次元物体のある平面で断面多角形を生成する第1のステップと、前記断面多角形の内部に代表点のボクセルを判別する第2のステップと、内部に代表点のあることが判別されたボクセルを用いて八分木を生成する第3のステップとを含む、八分木生成方法。

【請求項2】前記第2のステップは、前記代表点を通りかつ前記断面多角形の輪郭線と交差し、該代表点の一方側および他方側において前記輪郭線と交差する回数が奇数のとき前記代表点が内部にあり、偶数のとき前記代表点が外部にあると判別することを特徴とする、請求項1の八分木生成方法。

【請求項3】さらに、並列計算機を用いて、前記第1～第3のステップを処理することを特徴とする、八分木生成方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】この発明は八分木生成方法に関し、特に、三次元物体の多面体表現から八分木を生成するような八分木生成方法に関する。

## 【0002】

【従来の技術】三次元物体の形状データはさまざまな分野で用いられているが、近年は特に幅広い用途のアプリケーションにおいて、インタラクティブに効率よく処理することが求められるようになってきた。しかし、通常はある1つの形状表現法は限定された機能のみを持つため、その機能を最大限に利用できるような形状表現方法をアプリケーションに応じて注意深く選択する必要があった。たとえば、コンピュータグラフィックスを利用して写実的な画像を高速に生成するために、物体の多面体表現が用いられることが多い。

【0003】また、超二次関数や他のパラメトリックな方法による形状表現は、物体の変形を容易に表現することができ、またその関数による内外判定を利用することにより、物体間の干渉を高速に発見することができる。八分木もコンピュータビジョン、グラフィックス、ロボティクスなどの多くの分野で広く用いられている三次元表現法の1つである。

【0004】図9は八分木による物体の形状表現を説明するための図である。八分木による物体は、全体空間をルートノードとし、順次それを8分割された木で表現される。各ノードはホワイトノードとブラックノードにラベル付けされる。ホワイトノードは完全に物体の外部の空間を示し、ブラックノードは、完全に物体の内部の空間を示す。そうでない物体の内部と外部の両側にまたがる空間を示すノードは、グレイノードとされ、予め決め

られた最小の大きさに達するまで8つの子ノードに分割される。たとえば、図9(a)に示すようにルートノードは0, 1, 2…7にラベル付けされた1辺の長さが半分の子ノードに分割される。各々の子ノードは同じようにさらに分割される。図9(b)に示す物体の八分木表現を図9(c)に示す。

## 【0005】

【発明が解決しようとする課題】八分木は、三次元空間の階層的な表現法としての特徴を活かして、特定の部分空間を効率的にアクセスすることができるが、八分木で表現された物体形状を直接に生成したり、変形するのが困難であるという問題があった。

【0006】物体の多面体表現から八分木を表現する生成方法として、最も単純なものは、多面体の各面を含む平面で空間を分割し、各八分木ノードがそれらの内部にあるかどうかを判断するというものであるが、物体の形状は凸物体に限られるという欠点があった。

【0007】それゆえに、この発明の主たる目的は、非凸物体にも適用可能であって、多面体表現から高速に八分木を生成できるような八分木生成方法を提供することである。

## 【0008】

【課題を解決するための手段】請求項1に係る発明は、三次元物体の多面体表現から八分木を生成する八分木生成方法であって、三次元物体のある平面で断面多角形を生成する第1のステップと、断面多角形の内部に代表点のボクセルを判別する第2のステップと、内部に代表点のあることが判別されたボクセルを用いて八分木を生成する第3のステップとを含む。

【0009】請求項2に係る発明では、請求項1の第2のステップは、代表点を通りかつ断面多角形の輪郭線と交差し、その代表点の一方側および他方側において輪郭線と交差する回数が奇数のとき代表点が内部にあり、偶数のとき代表点が外部にあると判別する。

【0010】請求項3に係る発明では、並列計算機を用いて第1～第3のステップを処理する。

## 【0011】

【発明の実施の形態】図1はこの発明の一実施形態におけるボクセルの内点判定を説明するための図であり、図2は多角形の内点判定を説明するための図である。

【0012】図1において、物体1は面リストから構成される多面体表現として与えられるものとする。世界座標系の中で、物体が置かれている空間を  $M^3$ 、 $M = 2^{1+h}$  のボクセルに分割する。ここで、1は作成したい八分木の深さであり、hはある自然数である。これらの全ボクセルに対して、それぞれが物体の内部に存在するかどうかを判定し、これをもとに八分木を作成する。問題を単純化するため、ここでは各ボクセルの中心を代表点とし、この点の物体に対する内外の判定をもって、ボクセルの物体に対する内外を判断するものとする。

【0013】図1に示すように、与えられた多面体を代表点rを通る平面Qで切断し、そこに現れる断面多角形Pと点rの内外判定が行なわれる。図1の物体1を平面Qで切断したときの切断面は図2に示す断面多角形Pとなる。二次元平面内では、単純な多角形は平面の内部と外部の2つの互いに素な領域に分割する。これを利用すると、平面内で与えられた単純なn角形Pと点rに対して、点rがn角形Pの内部にあるかどうかはO(n)で判定できる。図2に示すように、点rを通る直線mを考えると、直線mがn角形Pの輪郭線と交差しない場合は、点rはn角形Pの外部にある。直線mがn角形Pと交差する場合は、直線mが点rの片側でn角形Pの輪郭線と交わる回数を数え、これらの回数が奇数のとき、点rはn角形Pの内部にあり、偶数のとき外部にあると判断できる。すなわち、図2に示した例では、直線m<sub>1</sub>はn角形Pの輪郭線点rの一方側でM<sub>r1</sub>～M<sub>r3</sub>で交差し、他方側でM<sub>r4</sub>～M<sub>r6</sub>で交差しており、それぞれ交差する回数Tは奇数であるため、点rはn角形Pの内部にあると判断できる。

【0014】一方、平面Q内で点sを通る直線Msはn角形Pの輪郭線と点sの一方側でMs<sub>1</sub>～Ms<sub>2</sub>で交差し、他方側でMs<sub>3</sub>～Ms<sub>4</sub>で交差しており、交差する回数Tは偶数であるため、点sはn角形Pの外部にあると判断できる。この点rが内部にあるボクセルをブラックボクセルとし、外部にあるボクセルをホワイトボクセルとする。

【0015】上述のごとく、各ボクセルの中心を代表点とし、この点の物体に対する内外の判定をもって、ボクセルの物体に対する内外を判断するようにしたが、これによれば、図3に示すように、ノードの一部が物体の内部にあっても、ホワイトボクセル3となる。しかし、アプリケーションによっては、このように一部が物体内部に占有されるリーフノードは、ブラックボクセル2となることが好ましい場合もある。たとえば、八分木で表現された物体同士の衝突を考える場合などは、物体形状は安全側に近似することが求められる場合もある。そこで、この発明では、図4に示すように、まず深さ1+hの八分木を作成し、その後八分木を辿って枝を刈り込むことにより、図5に示すような深さ1の八分木を作成する。そのため、発見されたブラックノードを空間全体をルートとする深さ1+hの八分木の形式で読み込む。そして、図6(a)に示すように8つの子ノードがすべてブラックボクセル2であればその親をブラックとし、図6(b)に示すようにこれらの子ノードを除く。また、深さ1+1以下のノードについては、図7(a)に示すように子ノードのうちの1つでもブラックボクセル2であればその親もブラックとして図7(b)に示すようにこれら子ノードを除く。これにより、基本アルゴリズムを独立した比較的単純な計算の繰り返しとして実現することができるので、並列処理による高速化の効果が期待できる。

る。

【0016】図8はこの発明の並列処理の概念を説明するための図である。一般に並列計算機では、プロセッサ間の通信手段の違いから、共有メモリを利用する方式と、メッセージパッシングの方式の2つの方式がある。前者の場合は、バスを経由してすべてのプロセッサから平等にアクセスできる共有メモリにデータを読み書きすることにより、プロセッサ間の通信を行なうので、時間が掛かると言われている。これに対して、後者の場合は、プロセッサごとに局所的なメモリ(分散メモリ)を持ち、必要なデータの送受を通信路を介して行なう。この発明ではこのような分散メモリを持つ並列計算機を用い、並列処理を行なう。その場合、2つの並列化の方法が考えられる。

【0017】第1の方法は、同一の断面内の複数のボクセルの内外判定を、複数のプロセッサで並行して実行しようとするものである。並行アルゴリズムを実行する前に、前処理として断面内のボクセルを全プロセッサに対して均等に分配する。すなわち、今、図8に示す断面Q<sub>k</sub>内の内外判定にN個のプロセッサが用いられたとすると、i番目のプロセッサは、i, i+N, i+2N, …番目のボクセルを受取り、処理することになる。分配された後は、各プロセッサは受取ったボクセルに対して上述の手順を実行し、そのボクセルが多角形P<sub>k</sub>の内部か外部かを判断する。断面Q<sub>k</sub>上のすべてのボクセルの内外判定が終了すると、次の断面Q<sub>k+1</sub>による断面多角形P<sub>k+1</sub>が生成され、同様に処理される。

【0018】並列化の第2の方法は、各断面にプロセッサを1つずつ割り当て、複数の断面内のボクセルの内外判定を、同じ数のプロセッサで並行して実行しようというものである。この場合、M個のプロセッサを使用してM枚の断面を同時に調べるものとすると、j番目のプロセッサはj, j+M, j+2M, …番目の断面内のボクセルに対して処理することになる。

【0019】これらの2つの方法を組合せて用いることももちろん考えられる。この場合、同一平面内のボクセルの内外判定にN個のプロセッサを用い、また同時にM個のプロセッサを用いてM枚の断面を同時に調べるとすると、合計M×N個のプロセッサが常時使われることになる。そして、断面Q<sub>k+p</sub>内のボクセルはpN+1, pN+2, …, (p+1)N番目のプロセッサ(0 < p < M)によって処理されることになる。

【0020】

【発明の効果】以上のように、この発明によれば、三次元物体のある平面で断面多角形のボクセルを生成し、そのボクセルの内部に代表点があるか否かを判別し、内部に代表点のあることが判別されたボクセルを用いて八分木を生成するようにしたので、並列計算機を用いて、非凸形状の物体の八分木表現を多面体表現から高速に作成することができる。

## 【図面の簡単な説明】

【図1】この発明の一実施の形態におけるボクセルの内点判定を説明するための図である。

【図2】多角形の内点判定を説明するための図である。

【図3】深さ1における八分木の近似方法を説明するための図である。

【図4】深さ1+hの八分木を示す図である。

【図5】深さ1+hの八分木を辿って枝を刈り込んだ状態を示す図である。

【図6】子ノードがすべてブラックボクセルのときに子ノードを除く状態を示す図である。

【図7】子ノードの1つでもブラックボクセルのときに、親をブラックボクセルとして子ノードを除く状態を示す図である。

示す図である。

【図8】この発明における並列処理の概念を説明するための図である。

【図9】八分木による物体の形状表現を説明するための図である。

## 【符号の説明】

1 物体

2 ブラックボクセル

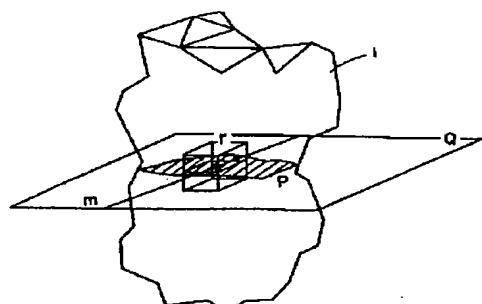
3 ホワイトボクセル

Q 平面

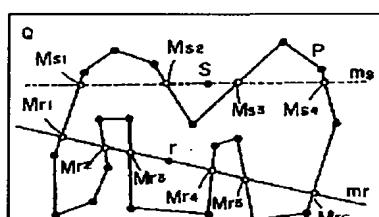
P 断面多角形

m 直線

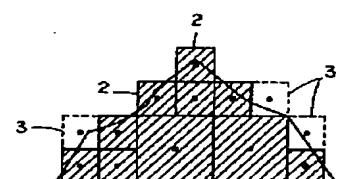
【図1】



【図2】

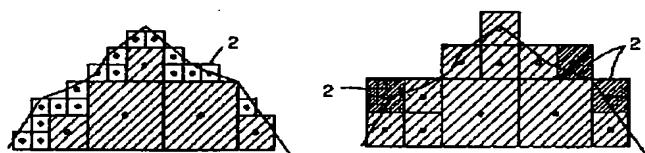


【図3】

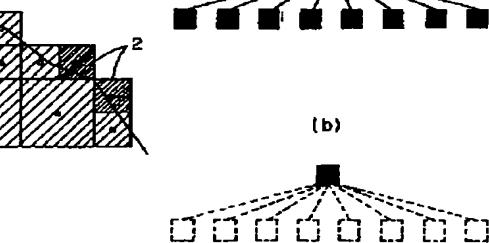


【図6】

【図4】

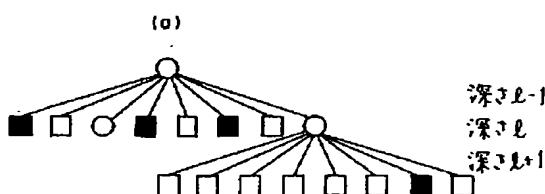


【図5】

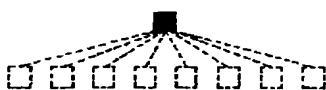


【図7】

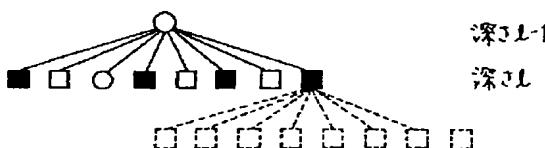
(a)



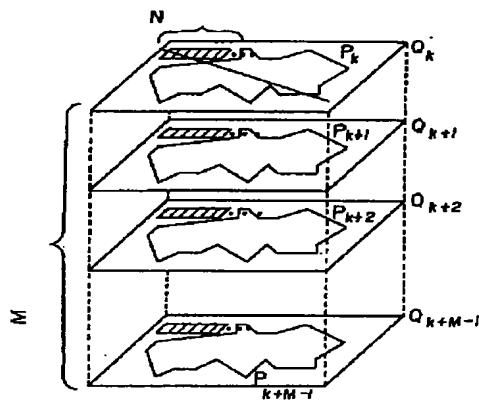
(b)



(b)



【図8】



【図9】

